CAPÍTULO 8

GERÊNCIA DO PROCESSADOR

Prezado(a) aluno(a),

Neste capítulo serão abordadas as funções básicas do escalonamento, políticas e algoritmos, e descritos os mecanismos implementados na gerência do processador.

É importante que você aprenda os escalonamentos Circulares e Prioridades, para que possa aprender o escalonamento Circular com Prioridades, muito utilizado na maioria dos sistemas operacionais de tempo compartilhado. Além disso, também é importante entender como o algoritmo Aging pode ser adicionado a esse escalonamento Circular com Prioridades para evitar Starvation.

Bom estudo!

Fala Professor

8.1. Introdução

Com o surgimento dos sistemas multiprogramáveis, em que múltiplos processos poderiam permanecer na memória principal, compartilhando o uso da CPU, a gerência do processador tornou-se uma das atividades mais importantes em um sistema operacional. A partir do momento em que diversos processos podem estar no estado de pronto, devem ser estabelecidos critérios para determinar qual processo está escolhido para fazer uso do processador. Os critérios utilizados para essa seleção compõem a chamada **política de escalonamento**, que é a base da gerência do processador e da multiprogramação em um sistema operacional.

8.2. Funções básicas

A política de escalonamento de um sistema operacional possui diversas funções básicas, como: manter o processador ocupado a maior parte do tempo, balancear o uso da CPU entre processos, privilegiar a execução de aplicações críticas, maximizar o *throughput* do sistema e oferecer tempos de resposta razoáveis para usuários interativos. Cada sistema operacional possui sua política de escalonamento adequada ao seu pro-

pósito e às suas características. Sistemas de tempo compartilhado, por exemplo, possuem requisitos de escalonamento distintos dos sistemas de tempo real.

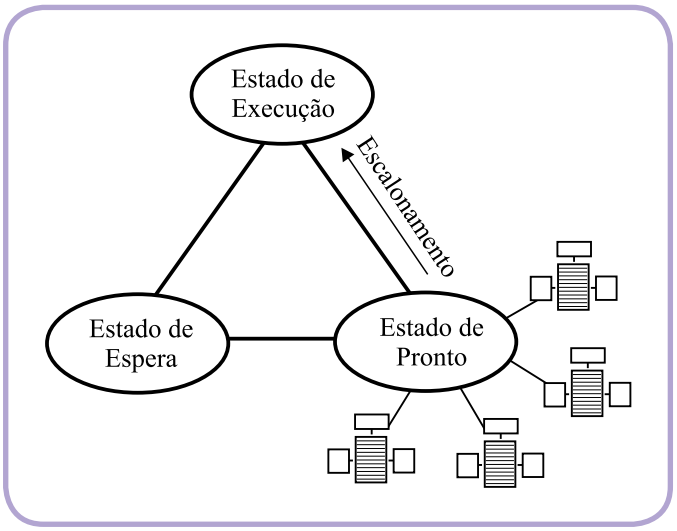


Figura 8-56: Escalonamento
Fonte: [1] – Machado e Maia, 2004. Adaptação.

A rotina do sistema operacional que tem como principal função implementar os critérios da política de escalonamento é denominada **escalonador** (*scheduler*). Em um sistema multiprogramável, o escalonador é fundamental, pois todo o escalonamento do processador é dependente dessa rotina.

Outra rotina importante na gerência do processador é conhecida como ***dispatcher***, responsável pela troca de contexto dos processos, após o escalonador determinar qual processo deve fazer uso do processador. O período de tempo gasto na substituição de um processo de execução é denominado latência de dispatcher.

8.3. Critérios de escalonamento

As características de cada sistema operacional determinam quais são os principais aspectos para a implementação de uma política de escalonamento adequada. Por exemplo, sistemas de tempo compartilhado exigem que o escalonamento trate todos os processos de forma igual, evitando assim, a ocorrência de *starvation*, ou seja, que um processo fique indefinidamente esperando pela utilização do processador. Já em sistemas de tempo real, o escalonamento deve priorizar a execução de processos críticos em detrimento da execução de outros processos.

A seguir, são apresentados os principais critérios que devem ser considerados em uma política de escalonamento:

- **Utilização do processador**

Na maioria dos sistemas, é desejável que o processador permaneça ocupado a maior parte do seu tempo.

- **Throughput**

Throughput representa o número de processos executados em um determinado intervalo de tempo. Quanto maior troughput, maior o número de tarefas executadas em função do tempo. A maximização do troughput é desejada na maioria dos sistemas.

- **Tempo de Processador / Tempo de CPU**

Tempo de processador ou tempo de CPU é o tempo que um processo leva no estado de execução durante seu processamento. As políticas de escalonamento não influenciam o tempo de processador de um processo, sendo este tempo função apenas do código da aplicação e da entrada de dados.

- **Tempo de Espera**

Tempo de espera é o tempo total que um processo permanece na fila de pronto durante seu processamento, aguardando para ser executado. A redução do tempo de espera dos processos é desejada pela maioria das políticas de escalonamento.

- **Tempo de *Turnaround***

Tempo de *turnaround* é o tempo que um processo leva, desde a sua criação até o seu término, levando em consideração todo o tempo gasto na espera, para a locação de memória, espera na fila de pronto (tempo de espera), processamento na CPU (tempo de processador) e no estado de bloqueado, como nas operações de E/S. As políticas de escalonamento buscam minimizar o tempo de *turnaround*.

- **Tempo de Resposta**

Tempo de resposta é o tempo decorrido entre uma requisição ao sistema ou à aplicação e o instante em que a resposta é exibida. Em sistemas interativos, pode-se entender tempo de resposta como tempo decorrido entre a última tecla digitada pelo usuário e o início da exibição do resultado no monitor. Em geral, o tempo de resposta não é limitado pela capacidade de processamento do sistema computacional, mas pela velocidade dos dispositivos de E/S.

De uma maneira geral, qualquer política de escalonamento busca otimizar a utilização do processador e o throughput, enquanto tenta diminuir os tempos de *turnaround*, espera e resposta. Apesar disso, as funções que uma política de escalonamento deve possuir são muitas vezes conflitantes. Dependendo do tipo de sistema operacional, um critério pode ter maior importância do que outros, como nos sistemas interativos onde o tempo de resposta tem grande relevância.

8.4. Escalonamento não-preemptivos e preemptivos

As políticas de escalonamento podem ser classificadas segundo a possibilidade de o sistema operacional interromper um processo em execução e substituí-lo por outro, atividade conhecida como **preempção**. Sistemas operacionais que implementam escalonamento com preempção são mais completos; porém, possibilitam políticas de escalonamento mais flexíveis.

O **escalonamento não-preemptivo** é aquele em que nenhum evento externo pode ocasionar a perda do uso do processador por um processo que está em execução. Assim, o processo somente sai do estado de execução caso termine seu processamento ou execute instruções do próprio código que ocasionem uma mudança para o estado de espera [1].

O escalonamento não-preemptivo foi o primeiro tipo de escalonamento implementado nos sistemas multiprogramáveis, em que predominava tipicamente o processamento batch.

O **escalonamento preemptivo** é caracterizado pela possibilidade de o sistema operacional interromper um processo de execução e passá-lo para o estado de pronto, com o objetivo de alocar outro processo na CPU [1].

Com o uso da preempção, é possível ao sistema, priorizar a execução de processos, como no caso de aplicações de tempo real, em que o fator tempo é crítico. Outro benefício é a possibilidade de implementar políticas de escalonamento que compartilhem o processador de uma maneira mais uniforme, distribuindo de forma balanceada o uso da CPU entre os processos.

Atualmente, a maioria dos sistemas operacionais possui políticas de escalonamento preemptivas que, apesar de tornarem os sistemas mais complexos, possibilitam a implementação dos diversos critérios de escalonamento apresentados.

Atividades

1. O que é política de escalonamento de um sistema operacional?
2. Quais as funções do escalonador e do dispatcher?
3. Quais os principais critérios utilizados em uma política de escalonamento?
4. Diferencie os tempos de processador, espera, turnaround e resposta.
5. Diferencie os escalonamentos preemptivos e não-preemptivos.

8.5. Escalonamento first-in-first-out (FIFO)

Conceitos

No **escalonamento First-In-First-Out** (***FIFO scheduling***), também conhecido como *first-come-first-served* (*FCFS scheduling*), o processo que chegar primeiro ao estado de pronto é o selecionado para a execução [1].

Esse algoritmo é bastante simples, sendo necessária apenas uma fila, onde os processos que passam para o estado de pronto entram no seu final e são escalonados quando chegam ao seu início. Quando o processo em

execução terminar seu processamento ou for para o estado de espera, o primeiro processo da fila de pronto será escalonado. Quando saem do estado de espera, todos os processos entram no final da fila de pronto.

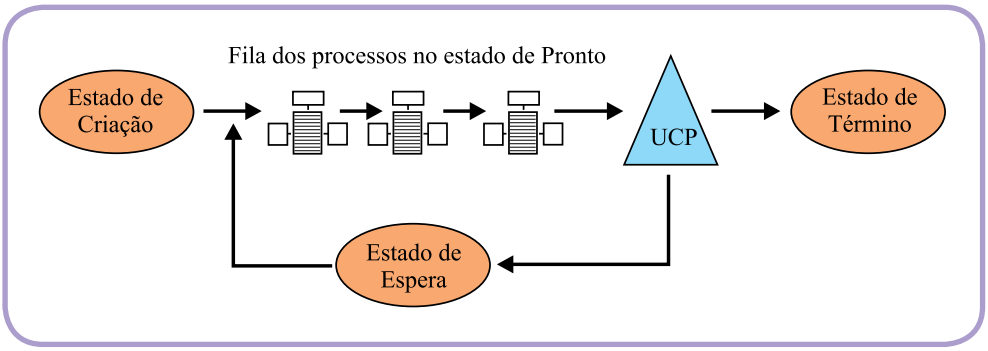
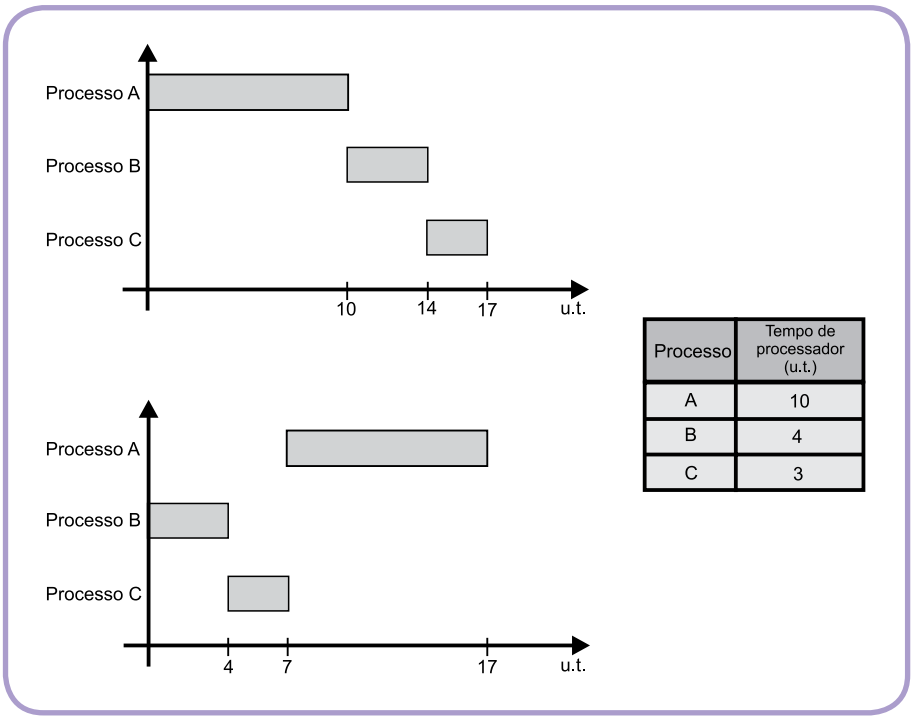


Figura 8-57: Escalonamento FIFO
Fonte: [1] – Machado e Maia, 2004. Adaptação.

Conceitos

Tempo média de espera na fila é a soma dos tempos de espera na fila de todos os processos, dividido pela quantidade de processos. A maioria dos algoritmos de escalonamento busca reduzir esse tempo [1, 2, 3, 4].

A Figura 8-58 apresenta dois exemplos do escalonamento FIFO, em duas situações distintas, em que os processos A, B e C são criados no instante de tempo 0, com os tempos de processador 10, 4 e 3, respectivamente. A diferença entre os exemplos apresentados na Figura 8-58 é o posicionamento dos processos na fila de pronto. O tempo médio de espera dos três processos no exemplo (a) é igual a (0+10+14)/3=8u.t., enquanto que no exemplo (b) é de aproximadamente (0+4+7)/3=3,7u.t. A diferença entre as médias é bastante significativa, justificada pela diferença dos tempos de processador entre os processos.



Processo	Tempo de processador (u.t.)
A	10
B	4
C	3

Figura 8-58: Exemplo de escalonamento FIFO
Fonte: [1] – Machado e Maia, 2004. Adaptação.

Apesar de simples, o escalonamento FIFO apresenta algumas deficiências. O principal problema é a impossibilidade de se prever quando um processo terá sua execução iniciada, já que isso varia em função do tempo de execução dos demais processos posicionados à sua frente na fila de pronto. O algoritmo de escalonamento não se preocupa em melhorar o tempo médio de espera dos processos, utilizando apenas a ordem de chegada dos processos à fila de pronto. Esse problema pode ser melhor percebido nos tempos de *turnaround* dos processos que demandam menor tempo de CPU.

Outro problema nesse tipo de escalonamento é que processos CPU-bound levam vantagem no uso do processador sobre processos I/O-bound. No caso de existirem processos I/O-bound mais importantes do que os CPU-bound, não é possível tratar esse tipo de diferença.

O escalonamento FIFO é do tipo não-preemptivo e foi inicialmente implementado em sistemas monoprogramáveis com processamento batch, sendo ineficiente, se aplicado na forma original em sistemas interativos de tempo compartilhado. Atualmente, sistemas de tempo compartilhado utilizam o escalonamento FIFO com variações, permitindo assim, parcialmente, sua implementação.

8.6. Escalonamento shortest-job-first (SJF)

Conceitos

No **escalonamento *Shortest-Job-First* (*SJF scheduling*)**, também conhecido como *shortest-process-next* (*SPN scheduling*), o algoritmo de escalonamento seleciona o processo que tiver o menor tempo de processador ainda por executar. Dessa forma, o processo em estado de pronto que necessitar de menos tempo de CPU para terminar seu processamento é selecionado para execução [1].

A Figura 8-59 mostra como ficaria o escalonamento, utilizando o algoritmo SJF, a partir dos mesmos processos utilizados no exemplo da Figura 8-58. O tempo médio de espera dos três processos, igual a (0+3+7)/3=3,3 u.t., é inferior aos tempos apresentados nos exemplos do escalonamento FIFO.

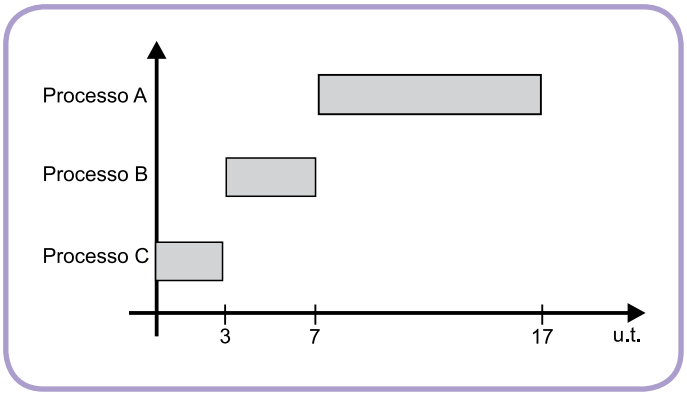


Figura 8-59: Exemplo de escalonamento SJF
Fonte: [1] – Machado e Maia, 2004. Adaptação.

O principal problema nessa implementação é a impossibilidade de estimar o tempo de processador para processos interativos, devido à entrada de dados ser uma ação imprevisível.

O escalonamento SJF, em sua concepção inicial, é um escalonamento não-preemptivo. Sua vantagem sobre o escalonamento FIFO está na redução do tempo médio de espera na fila pelos processos. Porém, para implementar o SJF, é necessário saber, de antemão, o tempo de execução de cada processo. Na prática, não é possível implementar o SJF da forma proposta, pois, para isso, o algoritmo teria que prever o futuro. Isso torna o SJF um algoritmo de escalonamento conceitual, pois suas ideias principais podem ser utilizadas em outros algoritmos de escalonamento.

Fala Professor

8.7. Escalonamento cooperativo

No **escalonamento cooperativo**, um processo em execução pode, voluntariamente, liberar o processador, retornando à fila de pronto, possibilitando que um novo processo seja escalonado, e permitindo, com isso, melhor distribuição no uso do processador [1].

Conceitos

A implementação desse escalonamento busca aumentar o grau de multiprogramação em políticas de escalonamento que não possuam mecanismo de preempção, como o FIFO e o SJF não-preemptivo.

A principal característica do escalonamento cooperativo está no fato de a liberação do processador ser uma tarefa realizada exclusivamente pelo processo em execução, que, de uma maneira cooperativa, libera a CPU para um outro processo. Nesse mecanismo, o processo em execução verifica periodicamente uma fila de mensagens, para determinar se existem outros processos na fila de pronto.

Como a interrupção do processo em execução não é responsabilidade do sistema operacional, algumas situações indesejadas podem ocorrer. No caso de um processo não liberar voluntariamente o processador, os demais não terão chance de serem executados até que o processo termine sua execução. Isso pode gerar sérios problemas para a multiprogramação cooperativa, na medida em que um programa pode permanecer um longo período alocando o processador.

Um exemplo desse tipo de escalonamento pode ser encontrado nos primeiros sistemas operacionais da família Microsoft Windows 3.11/95/98, sendo conhecido como multitarefa cooperativa.

8.8. Escalonamento circular

Conceitos

O **escalonamento circular** (***round robin scheduling***) é um escalonamento do tipo preemptivo, projetado especialmente para sistemas de tempo compartilhado. Esse algoritmo é bastante semelhante ao FIFO; porém, quando um processo passa para o estado de execução, existe um tempo limite para o uso contínuo do processador denominado **fatia de tempo** (***time-slice***) ou **quantum** [1, 2, 3, 4].

No escalonamento circular, toda vez que um processo é escalonado para a execução, uma nova fatia de tempo é concedida. Caso a fatia de tempo expire, o sistema operacional interrompe o processo em execução, salva seu contexto e direciona-o para o final da fila de pronto. Esse mecanismo é conhecido como **preempção por tempo** [1].

No escalonamento circular, a fila de processos em estado de pronto é tratada como uma fila circular. O escalonamento é realizado, alocando a CPU ao primeiro processo da fila de pronto. O processo permanecerá no estado de execução até que termine seu processamento, que voluntariamente passe para o estado de espera, ou que sua fatia de tempo expire, sofrendo, nesse caso, uma preempção pelo sistema operacional. Caso o processo sofra a preempção pelo sistema operacional, seu contexto será salvo e ele será colocado no final da fila de pronto. Após isso, um novo processo é escalonado com base na política de FIFO (o primeiro da fila será selecionado).

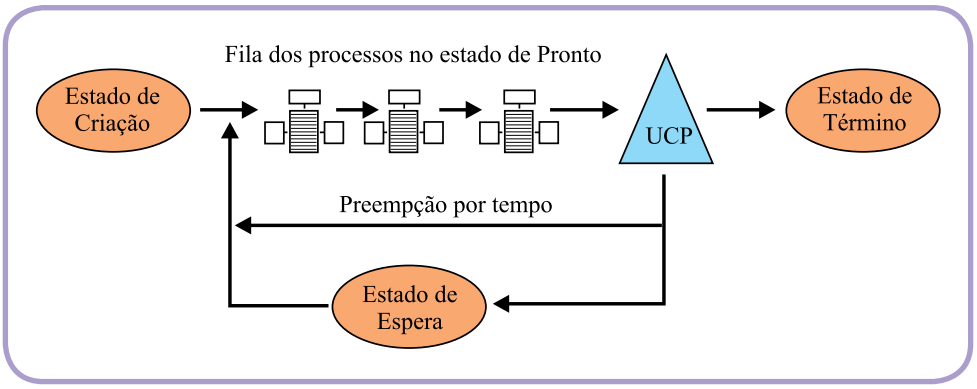


Figura 8-60: Escalonamento circular (***Round Robin***)
Fonte: [1] – Machado e Maia, 2004. Adaptação.

A Figura 8-61 exemplifica o escalonamento circular com três processos, em que a fatia de tempo é igual a 2u.t. Nesse exemplo não está sendo levado em consideração o tempo de latência do *dispatcher*, ou seja, o tempo de troca de contexto entre os processos.

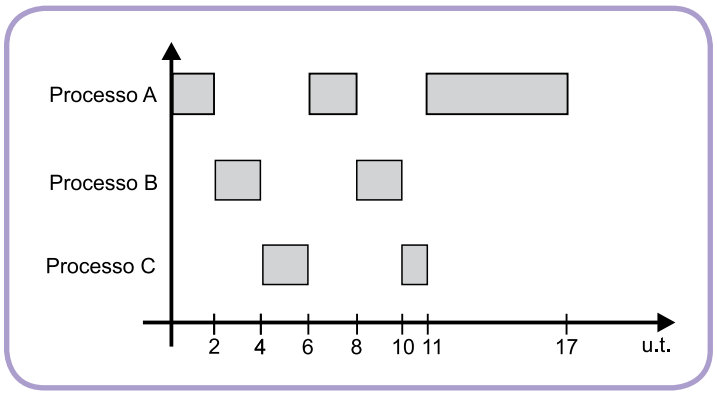


Figura 8-61: Exemplo de escalonamento circular (***Round Robin***)
Fonte: [1] – Machado e Maia, 2004. Adaptação.

O valor da fatia de tempo depende da arquitetura de cada sistema operacional e, em geral, varia entre 10 e 100 milissegundos (ms). Esse valor afeta diretamente o desempenho da política de escalonamento circular. Caso a fatia de tempo tenha um valor muito alto, esse escalonamento tenderá a ter o mesmo comportamento do escalonamento FIFO. Caso seja muito pequeno, a tendência é que haja um grande número de preempções, o que ocasionaria excessivas trocas de contexto, prejudicando o desempenho do sistema e afetando o tempo de turnaround dos processos.

Fala Professor

A principal vantagem do escalonamento circular é não permitir que um processo monopolize a CPU, sendo o tempo máximo alocado continuamente, igual à fatia de tempo definida no sistema. No caso de sistemas de tempo compartilhado, em que existem diversos processos interativos concorrendo pelo uso do processador, o escalonamento circular é adequado.

Um problema presente nessa política é que processos CPU-bound são beneficiados no uso do processador, em relação aos processos I/O-bound. Devido às suas características, os processos CPU-bound tendem a utilizar por completo a fatia de tempo enquanto os processos I/O-bound têm mais chances de passar para o estado de espera antes de sofrerem preempção por tempo. Essas características distintas ocasionam um balanceamento desigual no uso do processador entre os processos.

8.9. Escalonamento por prioridades

Conceitos

O **escalonamento por prioridades** é um escalonamento do tipo preemptivo, realizado com base em um valor associado a cada processo denominado **prioridade de execução**. O processo com maior prioridade no estado de pronto é sempre escolhido para execução, e processos com valores iguais são escalonados, seguindo o critério de FIFO. Nesse escalonamento, o conceito de fatia de tempo não existe; consequentemente, um processo em execução não pode sofrer preempção por tempo [1].

No escalonamento por prioridades, a perda do uso do processador só ocorrerá no caso de uma mudança voluntária para o estado de espera ou quando um processo de prioridade maior passa para o estado de pronto. Nesse caso, o sistema operacional deverá interromper o estado corrente, salvar seu contexto e colocá-lo num estado de pronto (mecanismo denominado **preempção por prioridade**). Após isso, o processo de maior prioridade é escalonado.

No escalonamento por prioridades, em cada prioridade existe uma fila de processos em estado de pronto que é tratada como uma fila circular. O escalonamento é realizado alocando o processador ao primeiro processo da fila de prioridade mais alta. O processo permanecerá no estado de execução até que termine seu processamento e, voluntariamente, passe para o estado de espera ou sofra uma preempção por prioridade.

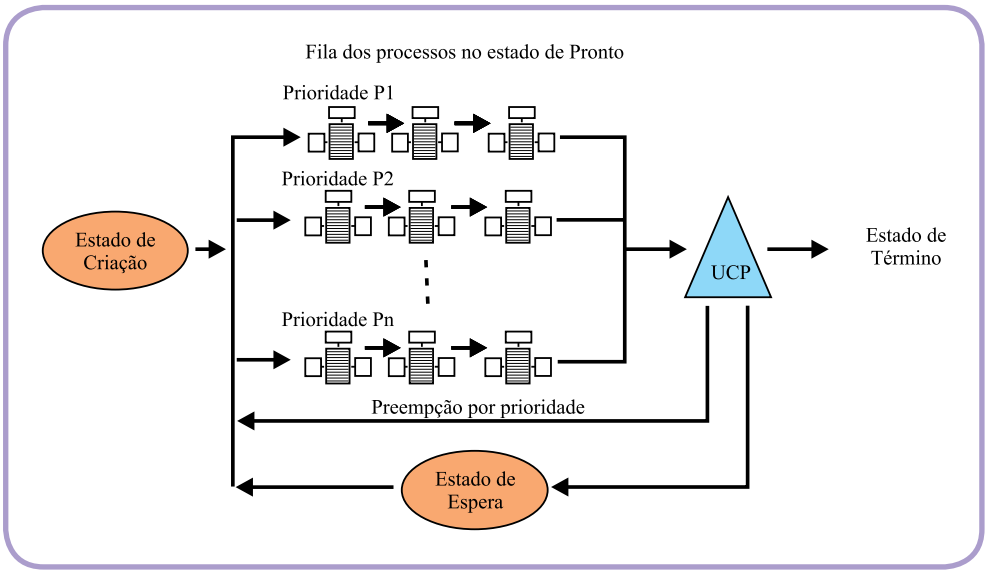
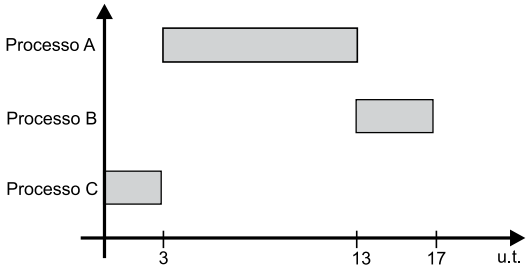


Figura 8-62: Escalonamento por prioridades
Fonte: [1] – Machado e Maia, 2004. Adaptação.

A Figura 8-63 exemplifica o escalonamento por prioridade de três processos de prioridades diferentes, em que o maior valor representa a prioridade mais alta.



Processo	Tempo de processador (u.t.)	Prioridade
A	10	2
B	4	1
C	3	3

Figura 8-63: Exemplo de escalonamento por prioridades
Fonte: [1] – Machado e Maia, 2004. Adaptação.

O escalonamento por prioridade também pode ser implementado de uma maneira não-preemptiva. Nesse caso, processos que passem para o estado de pronto com prioridade maior que a do processo em execução não ocasionam preempção, sendo apenas colocadas no início da fila de pronto.

Cada sistema operacional implementa sua faixa de valores para as prioridades de execução. Alguns sistemas associam as maiores prioridades a valores altos, enquanto outros sistemas utilizam valores baixos. No caso do OpenVMS, a prioridade do processo pode variar de 0 a 31, sendo 31 a maior prioridade. No IBM-AIX, a prioridade varia de 0 a 127, porém os valores mais baixos possuem maior prioridade de execução.

Fala Professor

A prioridade de execução é uma característica do contexto do software de um processo e pode ser classificada como estática ou dinâmica. A **prioridade estática** não tem o seu valor alterado durante a existência do processo, já a **prioridade dinâmica** pode ser ajustada de acordo com critérios definidos pelo sistema operacional. A possibilidade de alterar o valor da prioridade de um processo ao longo de seu processamento permite ajustar o critério de escalonamento em função do comportamento de cada processo no sistema.

8.10. Escalonamento circular com prioridades

Conceitos

O **escalonamento circular com prioridades** implementa o conceito de fatia de tempo e de prioridade de execução, associada a cada processo. Nesse tipo de escalonamento, um processo permanece num estado de execução até que termine seu processamento e, voluntariamente, passe o estado de espera ou sofra uma preempção por tempo ou prioridade [1, 2, 3, 4].

A principal vantagem desse escalonamento é permitir o maior balanceamento no uso da CPU, com a possibilidade de diferenciar o grau de importância dos processos. Esse tipo de escalonamento é amplamente utilizado em sistema de tempo compartilhado.

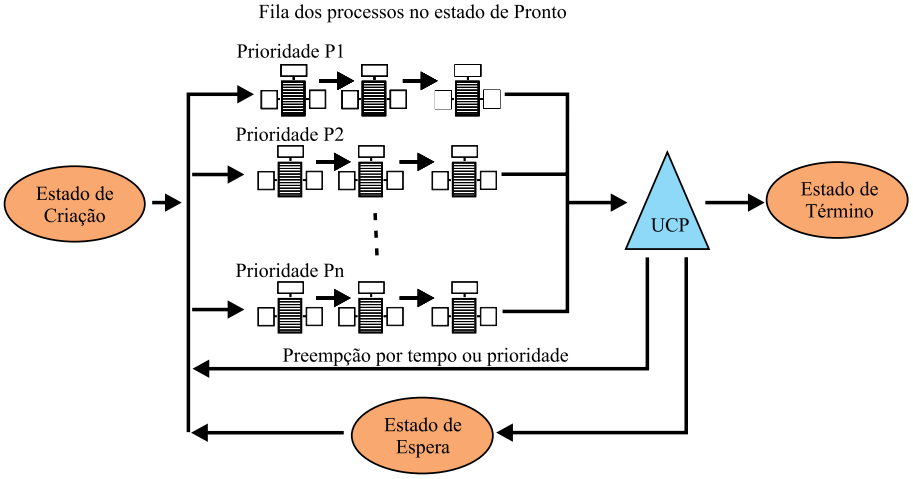


Figura 8-64: Escalonamento circular com prioridades
Fonte: [1] – Machado e Maia, 2004. Adaptação.

Um dos principais problemas no escalonamento por prioridades é o **starvation**. Processos de baixa prioridade podem não ser escalonados, permanecendo indefinidamente na fila de pronto. Uma solução para esse problema, possível em sistemas que implementam prioridade dinâmica, é a técnica de **aging**. Por meio desse mecanismo, um processo com baixa prioridade tem sua prioridade aumentada, gradativamente, com o passar do tempo, até que tenha prioridade suficiente para executar. Após executar durante sua fatia de tempo, o processo volta à fila inicial, com menor prioridade, para que tenha sua prioridade novamente aumentada, também, com o passar do tempo. Isso faz muito sentido, pois se um processo tem baixa prioridade, pressupõe-se que ele executará menos frequentemente que os processos de maior prioridade.

O escalonamento por prioridades possibilita diferenciar os processos segundo critérios de importância. Com isso, processos de maior prioridade são escalonados preferencialmente. Isso é bastante útil tanto em sistemas de tempo real e nas aplicações de controle de processo, como nas aplicações de sistemas de tempo compartilhado, em que, às vezes, é necessário priorizar o escalonamento de determinados processos.

8.11. Escalonamento por múltiplas filas

Conceitos

No **escalonamento por múltiplas filas** (***multilevel queue scheduling***) existem diversas filas de processos no estado de pronto, cada qual com uma prioridade específica. Os processos são associados às filas em função de características próprias, como importância para a aplicação, tipo de processamento ou área de memória necessária [1].

Como os processos possuem características de processamento distintas, é difícil que um único mecanismo de escalonamento seja adequado a todos. A principal vantagem de múltiplas filas é a possibilidade de convivência de mecanismos de escalonamento distintos em um mesmo sistema operacional. Cada fila possui um mecanismo próprio, permitindo que alguns processos sejam escalonados pelo mecanismo FIFO, enquanto outros, pelo circular.

Nesse mecanismo, o processo não possui prioridade, ficando essa característica associada à fila. O processo em execução sofre preempção,

caso um outro processo entre em uma fila de maior prioridade. O sistema operacional só pode escalonar processos de uma determinada fila, caso todas as outras filas, de maior prioridade, estejam vazias. Uma boa prática é classificar os processos em função do tipo de processamento realizado e associá-los adequadamente às respectivas filas.

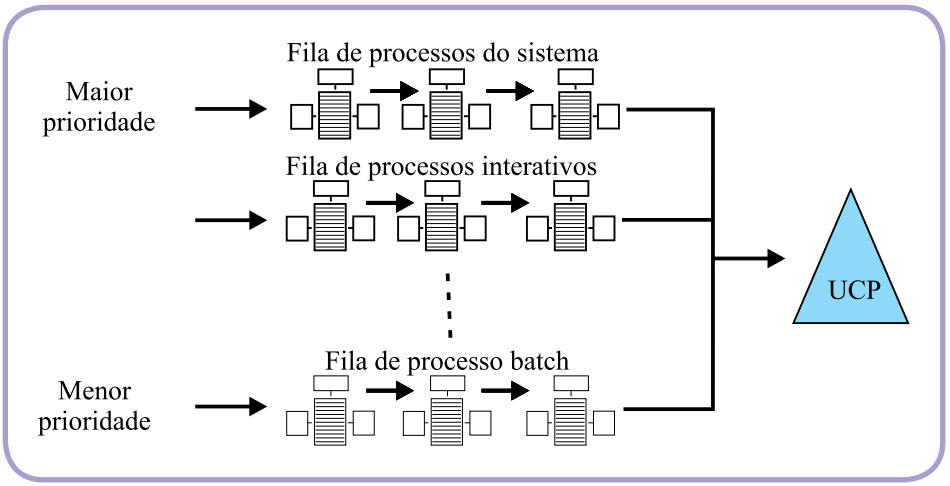


Figura 8-65: Escalonamento por múltiplas filas
Fonte: [1] – Machado e Maia, 2004. Adaptação.

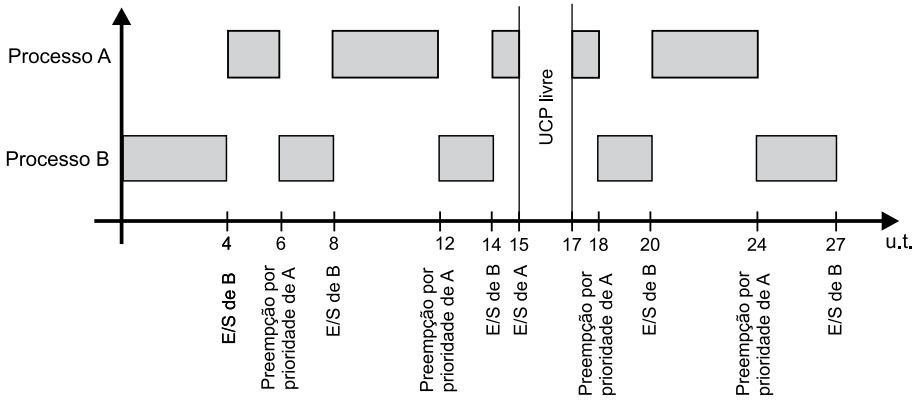
Uma desvantagem deste escalonamento é que, no caso de um processo alterar seu comportamento no decorrer do tempo, o processo não poderá ser redirecionado para uma outra fila mais adequada. A associação de um processo à fila é determinada na criação do processo, permanecendo até o término do seu processamento.

8.12. Política de escalonamento em sistemas de tempo compartilhado

Em geral, sistemas de tempo compartilhado caracterizam-se pelo processamento interativo, em que usuários interagem com as aplicações, exigindo tempos de respostas baixos. A escolha de uma política de escalonamento para atingir esse propósito deve levar em consideração o compartilhamento dos recursos, de forma equitativa, para possibilitar o uso balanceado da CPU entre processos [1].

Um refinamento no balanceamento e uso do processador pode ser obtido implementando-se o escalonamento circular com prioridades dinâmicas. Com isso, é possível ao administrador do sistema alterar a prioridade de um processo em função do uso excessivo ou reduzido do uso do processador. Outro benefício é que alguns sistemas podem alterar dinamicamente a prioridade dos processos em função do tipo de operação de E/S realizada. Algumas operações de E/S são mais len-

tas, fazendo com que um processo permaneça mais tempo no estado de espera sem chances de competir pelo uso do processador. Nesse caso, quando o processo sai do estado de espera, um acréscimo temporário à sua prioridade é concedido pelo sistema operacional. Dessa forma, os processos têm maiores chances de serem escalonados, compensando parcialmente o tempo gasto no estado de espera. É importante perceber que os processos CPU-bound não são prejudicados com essa política, pois podem ser executados durante o período em que os processos I/O-bound estão num estado de espera.



Processo	Tempo de processador (u.t.)	Característica	Prioridade
A	12	CPU - bound	Baixa
B	13	I/O- bound	Alta

Figura 8-66: Exemplo de escalonamento circular com prioridades
Fonte: [1] – Machado e Maia, 2004. Adaptação.

Embora os sistemas com prioridade dinâmica sejam mais complexos de implementar que os sistemas com prioridade estática, o tempo de resposta oferecido compensa. Atualmente, a maioria dos sistemas operacionais de tempo compartilhado utiliza o escalonamento circular com prioridades dinâmicas.

8.13. Política de escalonamento em sistemas de tempo real

Diferentemente dos sistemas de tempo compartilhado, em que a aplicação não é prejudicada pela variação no tempo de resposta, algumas aplicações específicas exigem respostas imediatas para a execução de determinadas tarefas. Nesse caso, a aplicação deve ser executada em sistemas operacionais de tempo real, em que é garantida a execução de processos dentro de limites rígidos de tempo, sem o risco de a aplicação

ficar comprometida. Aplicações de controle de processos como sistemas de controle de produção de bens industriais e controle de tráfego aéreo são exemplos de aplicação de tempo real.

O escalonamento em sistemas de tempo real deve levar em consideração a importância relativa de cada tarefa na aplicação. Em função disso, o escalonamento por prioridades é o mais adequado, já que para cada processo uma prioridade é associada em função da importância do processo dentro da aplicação. No escalonamento para sistemas de tempo real não deve existir o conceito de fatia de tempo e a prioridade de cada processo deve ser estática.

Indicações

[1] MACHADO, F.B. e MAIA, L.P. ***Arquitetura de Sistemas Operacionais***. 4.ed. LTC, 2007.

[2] SILBERSCHATZ, A., GALVIN, P.B., GAGNE, G. ***Fundamentos de Sistemas Operacionais***. 6.ed. LTC, 2004.

[3] TANENBAUM, A.S. ***Sistemas Operacionais Modernos***. 2.ed. Pearson Brasil, 2007.

[4] OLIVEIRA, R.S., CARISSIMI, A.S., TOSCANI, S.S. ***Sistemas Operacionais***. 3.ed. Sagra-Luzzato. 2004.

Atividades

Atividades

1. Qual a diferença entre os escalonamentos FIFO e circular?
2. Descreva o escalonamento SJF e o escalonamento por prioridades.
3. Qual a diferença entre preempção por tempo e preempção por prioridade?
4. O que é um mecanismo de escalonamento adaptativo?
5. Que tipo de escalonamento aplicações de tempo real exigem?
6. O escalonamento por múltiplas filas com realimentação, favorece processos CPU-bound ou I/O-bound? Justifique.
7. Considere que cinco processos sejam criados no instante de tempo 0 (P1 , P2 , P3 , P4 e P5) e possuam as características descritas na tabela a seguir:

ProcessoTempo de CPUPrioridade

Atividades

P1103P2144P351P472P5205 Desenhe um diagrama ilustrando o escalonamento dos processos e seus respectivos tempos de turnaround, segundo as políticas especificadas a seguir. O tempo de troca de contexto deve ser desconsiderado.

a) FIFO;

b) SJF;

c) Prioridade (número menor implica prioridade maior) ;

d) Circular com fatia de tempo igual a 2 u.t.

8. Considere um sistema operacional com escalonamento por prioridades onde a avaliação do escalonamento é realizada em um intervalo mínimo de 5ms. Neste sistema, os processos A e B competem por uma única CPU. Desprezando os tempos de processamento relativo às funções do sistema operacional, a tabela a seguir fornece os estados dos processos A e B ao longo do tempo, medido em intervalos de 5 ms (E=execução, P=pronto e W=espera). O processo A tem menor prioridade que o processo B.

	50-54	55-59	60-64	65-69	70-74	75-79	80-84	85-89	90-94	95-99	100 - 105
PROCESSO A	P	E	P	P	E	E	W	W	P	E	E
PROCESSO B	W	P	E	E	W	W	P	E	E	-	-

a) Em que tempos A sofre preempção?

b) Em que tempos B sofre preempção?

c) Refaça a tabela anterior supondo que o processo A é mais prioritário que o processo B.

9. Como o valor do quantum pode afetar o grau de multiprogramação em um sistema operacional? Qual a principal desvantagem de um quantum com um valor muito pequeno?

Atividades

10. Considere um sistema operacional que implemente escalonamento circular com fatia de tempo igual a 10 u.t.. Em um determinado instante de tempo, existem apenas três processos (P1, P2 e P3) na fila de pronto, e o tempo de CPU de cada processo é 18, 4 e 13 u.t, respectivamente. Qual o estado de cada processo no instante de tempo T, considerando a execução dos processos P1, P2 e P3, nesta ordem, e que nenhuma operação de E/S é realizada?

    a) T = 8 u.t.

    b) T = 11 u.t.

    c) T = 33 u.t.

11. Considere um sistema operacional que implemente escalonamento circular com fatia de tempo igual a 10 u.t. Em um determinado instante de tempo, existem apenas três processos (P1, P2 e P3) na fila de pronto, e o tempo de CPU de cada processo é 14, 4 e 12 u.t, respectivamente. Qual o estado de cada processo no instante de tempo T, considerando a execução dos processos P1, P2 e P3, nesta ordem, e que apenas o processo P1 realiza operações de E/S? Cada operação de E/S é executada após 5 u.t. e consome 10 u.t.

    a) T = 8 u.t.

    b) T = 18 u.t.

    c) T = 28 u.t.

12. Existem quatro processos (P1, P2, P3 e P4) na fila de pronto, com tempos de CPU estimados em 9, 6, 3 e 5, respectivamente. Em que ordem os processos devem ser executados para minimizar o tempo de turnaround dos processos?

13. Considere a tabela a seguir onde:

Processo	Tempo de CPU	Prioridade
P1	4	0
P2	2	0
P3	5	1
P4	3	0

Atividades

Qual o tempo de turnaround médio dos processos considerando o tempo de troca de contexto igual a 0 e a 5 u.t. para os seguintes escalonamentos:

a) FCFS;

b) SJF;

Circular com fatia de tempo igual a 20 u.t.